

Univerza v Ljubljani  
Fakulteta za računalništvo in informatiko

Marko Žankar

# **Razvoj aplikacije za arhiviranje dokumentov**

DIPLOMSKO DELO  
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Rok Rupnik

Ljubljana, 2013



Št. naloge: 00507 / 2013  
Datum: 15.4.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MARKO ŽANKAR**

Naslov: **RAZVOJ APLIKACIJE ZA ARHIVIRANJE DOKUMENTOV**  
**THE DEVELOPMENT OF APPLICATION FOR DOCUMENT ARCHIVING**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Osnovni namen orodij širšega področja arhiviranja dokumentov je omogočiti obdelavo gradiv: zajem dokumentov iz zunanjih informacijskih sistemov, morebitno proženje delovnih procesov na podlagi vsebine dokumentov ter po zaključku procesov arhiviranje teh dokumentov v sistem za elektronsko hrambo dokumentov. Razvoj in vzdrževanje takšnih namenskih aplikacij predstavlja visoke stroške vzdrževanja in administriranja, saj so praviloma v takšni verigi tri do štiri aplikacije. Zasnуйте, izdelajte načrt in razvite aplikacijo, ki bo obsegala funkcionalnosti celotne zgoraj omenjene verige. Aplikacija naj uporabniku omogoča enostavno pripravo nove obdelave gradiva brez programiranja. Aplikacija naj uporablja programske vtičnike, ki so razdeljenimi v kategorije separacije, validacije, pretvorbe in izhodnih modulov. Arhitektura aplikacije naj omogoča, da se novi vtičniki lahko razvijajo ločeno od osnovne aplikacije.

Mentor:

doc. dr. Rok Rupnik



Dekan:

prof. dr. Nikolaj Zimic

## Izjava o avtorstvu diplomskega dela

Spodaj podpisani Marko Žankar, z vpisno številko 63050375, sem avtor diplomskega dela z naslovom:

*Razvoj aplikacije za arhiviranje dokumentov*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 22. Oktobra 2013

Podpis avtorja:

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

# Povzetek

V diplomski nalogi je predstavljen razvoj aplikacijske rešitve, ki je nadomestila številna namensko izdelana orodja za zajem in pretvorbo dokumentov v digitalni obliki oziroma obdelavo elektronskih dokumentov. Osnovni namen orodij, katere je aplikacijska rešitev nadomestila, je bil zajem dokumentov iz eksternih informacijskih sistemov, proženje delovnih procesov (proženje informatiziranih delovnih procesov na različnih BPM platformah) ter po zaključku procesov arhiviranje teh dokumentov v sistem za elektronsko hrambo dokumentov. Razvoj in vzdrževanje namenskih aplikacijskih rešitev, ki so bile v uporabi, je zaradi njihove številčnosti predstavljal visoke stroške vzdrževanja in administriranja. Cilj razvoja nove aplikacijske rešitve je bil, razviti enotno rešitev s takšnim naborom funkcionalnosti, ki bi pokrila večino potrebnih funkcionalnosti in bi tehnologu ali administratorju omogočala enostavno pripravo nove obdelave gradiva, ki ne bo zahtevala posebnega dodatnega znanja o razvoju programske opreme in programiranju. Aplikacijska rešitev, kot končni izdelek, je pripravljena strežniška aplikacija, ki za obdelavo uporablja programske vtičnike, ki nudijo vse funkcionalnosti, ki so jih nudile prehodno uporabljene programske rešitve. Aplikacijska rešitev s programskimi vtičniki, razdeljenimi na posamične gradnike omogoča pripravo različnih vrst obdelav gradiva. Vtičniki oz. gradniki so razdeljeni v kategorijo separacije, validacije, pretvorbe in izhodnih modulov. Novi gradniki se z uporabo pripravljenega API vmesnika lahko razvijajo ločeno od matične strežniške aplikacije.

## **Ključne besede:**

aplikacijska rešitev, arhiviranje, pretvorba in zajem dokumentov, brezpapirno poslovanje, programske vtičniki.

# Abstract

The thesis presents a development of software application solution, which solves the problem of many purpose built electronic document capture and conversion tools. The primary aim of those tools was: document capture from an external information system, triggering workflows (triggering computerised workflow processes in different BPM platforms) and archiving documents in to electronic archive systems. Over time these solutions were starting to pile up, which resided in a costly development and maintenance. The objective was to develop a uniform solution that would cover most of the current functionality and enable preparation of a new document processing job, to a human resource without knowledge of programming a new solution. The final product has been devised for this purpose, as a server application which uses software plugins to add all the functionality of old tools separated into single programme blocks that can be used for every processing job. Plugins itself are categorized to the following categories: Separation, validation, conversion and release plugins. The plugins can be developed separately from the main server application, through the use of the specified API interface.

## **Key words:**

Application solution, Archiving, Document Capture and conversion, Paperless business operations, software plugins.

# Vsebina

<b>Povzetek.....</b>	<b>V</b>
<b>Abstract .....</b>	<b>VI</b>
<b>1    Uvod .....</b>	<b>1</b>
1.1    Namen naloge .....	2
1.2    Cilji naloge .....	3
1.3    Uporabljene kratice.....	3
<b>2    Zakonske zahteve in zahteve standardov .....</b>	<b>6</b>
2.1    Zakonodaja s področja zajema in pretvorbe dokumentarnega gradiva .....	6
2.2    Standardi s področja zajema in pretvorbe dokumentarnega gradiva .....	7
2.2.1    Splošne zahteve oblike zapisa za dolgoročno hrambo .....	7
2.2.2    Najpogostejše uporabljene oblike zapisa.....	8
2.2.3    PDF/A – Oblika zapisa.....	10
2.2.4    TIFF – Oblika zapisa.....	11
2.2.5    WC3 XML – Oblika zapisa .....	13
2.2.6    Dovoljeni standardi kompresije.....	13
2.3    Zahteva za metapodatkovno shemo .....	15
2.3.1    Opredelitev in namen.....	15
2.3.2    Minimalne zahteve.....	16
<b>3    Razvoj programske opreme.....</b>	<b>19</b>
3.1    Predstavitev metodologije - Sashimi Waterfall Model.....	19
3.1.1    Waterfall model .....	19
3.1.2    Sashimi Model .....	19
3.1.3    Faze Sashimi Waterfall Model modela.....	20
3.2    Razvoj.....	22
3.2.1    Zasnova programske rešitve.....	22
3.2.2    Funkcionalna specifikacija.....	23
3.2.3    Opis najbolj pogosto uporabljenih vtičnikov .....	28
<b>4    Primeri praktične uporabe .....</b>	<b>32</b>
4.1    Preproste obdelave.....	32
4.2    Primeri zahtevnih obdelav .....	33
4.2.1    Prejeti elektronski računi .....	33
4.2.2    PDF izpiski .....	36

4.2.3	<i>Obdelava in priprava izdanih računov.....</i>	39
<b>5</b>	<b>Sklepne ugotovitve.....</b>	<b>41</b>
<b>6</b>	<b>Literatura .....</b>	<b>42</b>



# 1 Uvod

Poslovni subjekti želijo poslovati s čim manjšimi stroški z namenom izboljševanja svoje konkurenčnosti na trgu. Zmanjševanje stroškov je mogoče doseči z optimizacijo poslovnih procesov. V okviru optimizacije poslovnih procesov se analizirajo obstoječi procesi z iskanjem možnosti za njihovo optimizacijo. To pomeni, da želijo procese pohitriti, zmanjšati potrebne čase za njihovo izvedbo in omogočiti zaposlenim enostavnejše in hitreje opravljanje dela ter jim omogočiti dostop do različnih informacij in podatkov z enega mesta, če je mogoče iz aplikacije, ki jo vsakodnevno uporabljajo pri svojem delu. V okviru optimizacij poslovnih procesov se poslovni subjekti nemalokrat odločajo za informatizacijo teh procesov. V okviru teh, se pojavi potreba po uporabi dokumentov v digitalni obliki, katere je mogoče distribuirati skozi informatizirane procese. Z uporabo dokumentov v digitalni obliki pa se zmanjšajo oziroma v celoti odpravijo stroški, ki nastanejo z tiskanjem, kopiranjem, multipliciranjem teh dokumentov. Z uporabo digitaliziranih dokumentov in z informatizacijo procesa zagotovimo, da obstaja v sistemu samo ena en izvod dokumenta, ki je na voljo več uporabnikom, prav tako pa se skrajša čas distribucije dokumenta do končnega uporabnika. Da pa lahko to zagotovimo je potrebno omogočiti zajem dokumentov iz zalednih sistemov podjetja, bodisi za potrebe proženja informatiziranih poslovnih procesov kot za potrebe hrambe dokumentov v sistemih za elektronsko hrambo gradiva. Nemalokrat bi zajem dokumentov predstavljal potrebo po nadgradnji zalednih aplikacij, kar običajno predstavlja velike stroške. Z uporabo namenske aplikacije, katero smo razvili, pa to ni potrebno ali pa so posegi v zaledne aplikacije minimalni. Mnogi sistemi že omogočajo zajem dokumentov, tako imenovani COLD izvoz, ki pa ne omogoča proste distribucije dokumentov uporabnikom. Namesto tega se lahko uporabi ciljna pretvorba, ki služi vizualizaciji dokumenta, ki je enostavno berljiv in služi kot vhod v delovni proces (informatiziran delovni proces), to je distribuciji dokumenta po elektronski poti drugim uporabnikom. Pri tem želimo zagotoviti sledeče:

- prikaz dokumentov v zalednih sistemih, neposredno iz sistema za hrambo dokumentov (prikaz na zahtevo),
- možnost proženja delovnih procesov v sistemih za obvladovanje delovnih procesov (BPM, ERP, ki ima vgrajeno podporo delovnim procesom),
- delo z končnimi elektronskimi dokumenti v večini pomeni samo arhiviranje, posebej v primerih, kjer dokumenti niso procesno pomembni, kot so izpiski ali opomini. Za prikaz le teh v zalednih sistemih uporabijo integracijo z arhivom, ki jim služi kot neposreden priklic dokumentov na zahtevo.

Tak primer zajema dokumentov predstavlja avtomatska likvidacije elektronskih računov, katerega bomo predstavili v nadaljevanju.

Glavne težave oziroma omejitve pri pripravi aplikacije, ki bo omogočala pretvorbo in zajem (uvoz) gradiva v ciljne sisteme, predstavljajo različni mednarodni standardi ali dogovorjeni standardi med različnimi poslovnimi subjekti, kot je primer pri elektronskih računih, kjer se upošteva standard E-sloga 1.6, ZBS shema za prenos podatkov o sodnih sklepih ali pa samo določena dogovorjena sestava glede na informacijski sistem v katerega se zajema. Pri pripravi rešitve pa je potrebno upoštevati tudi zakonodajne zahteve glede formatov zapisa za hrambo gradiva v digitalni obliki, ki so primerni elektronsko hrambo dokumentov, posebej za dolgoročno hrambo dokumentov v elektronski obliki (hramba daljša od 5 let), saj želimo zagotoviti primerne formati zapisa, ki ne bodo zahtevali dodatne pretvorbe formata zapisa pred zajemom v sistem za elektronsko hrambo.

Pri gradnji aplikacijske rešitve smo bili dolžni upoštevati vse te zahteve. Z povečanim povpraševanjem za zajem različnih tipov dokumentov se je povečevala količina namensko uporabljenih aplikacij, kar pa je privedlo do oteženega vzdrževanja in obvladovanja sprememb.

V tej nalogi je predstavljen razvoj rešitve, s katero smo lahko zagotovili delovanje le te, skladno z vsemi zakonskimi zahtevami, hkrati pa smo z inovativno rešitvijo odpravili potrebo po namensko grajenih aplikacijah.

## **1.1 Namen naloge**

Namen diplomske naloge je predstaviti aplikacijsko rešitev za obdelavo in zajem elektronskih dokumentov. Pri načrtovanju in razvoju aplikacijske rešitve smo upoštevali tako cikel razvoja programske opreme kakor tudi vse zakonske zahteve s področja obdelave in zajema dokumentov v elektronski obliki. Poleg navedenega smo pri načrtovanju morali upoštevati tudi funkcionalne zahteve. Funkcionalne zahteve so izhajale iz funkcionalnosti, ki so jih nudile predhodno uporabljane namenske aplikacije. Zahtevane funkcionalne zahteve smo razdelili na manjše gradnike, s čimer smo zagotovili čim večjo vsestranskost in skalabilnost nove aplikacije.

Odločitev za izdelavo tovrstne aplikacije je bila potrebna zaradi velikega števila formatov zapisa elektronskih dokumentov, ki nastajajo v informacijskih sistemih in jih morajo biti aplikacije sposobne zajeti. Z razvojem aplikacijske rešitve smo razvili aplikacijo, ki omogoča delo za raznovrstnimi formati zapisa, s čimer pa bo končnim uporabnikom omogočen zajem elektronskih dokumentov ter pretvorba let teh v primerne formate brez dodatnih stroškov lastnega razvoja in vzdrževanja ali prilaganja zalednih informacijskih sistemov.

Diplomsko nalogo sestavljajo trije sklopi ali deli. V prvem delu vam bomo v predstavili področje zajema in pretvorbe dokumentov in zakonske zahteve Arhiva Republike Slovenije (v nadaljevanju Arhiv RS), v katerem bomo predstavili teoretična izhodišča in osnovo, ki jo je potrebno upoštevati pri načrtovanju aplikacij s tega področja. V drugem delu se bomo osredotočili na cikel razvoja programske opreme in same funkcionalne zahteve za aplikacijo. V tem delu vam bomo približali strukturo aplikacije in metodologijo razvoja. V tretjem delu pa vam bomo prikazali nekaj splošnih primerov uporabe aplikacije, ki so najbolj pogosti, s katerimi bomo lažje prikazali delovanje aplikacije in njeno uporabno vrednost.

## 1.2 Cilji naloge

Cilji in naloge aplikacije za zajem elektronskih dokumentov:

- Prezem različnih tipov datotek.
- Prezem različnih standardov zapisov datotek.
- Zagotovitev pravilne pretvorbe gradiva, ki ustreza Enotnim tehnološkim zahtevam (ETZ).
- Zajem dokumenta in zajem metapodatkov z dokumenta po vnaprej dogovorjenem standardu z namenom pravilnega zajema (izvoza) v ciljni sistem (DMS, poljuben zaledni sistem ali sistem za e-hrambo).
- Sistematično sledenje in sporočanje napak s katerim zagotovimo možnost kontrole nad zajetimi dokumenti in metapodatki, možnost izvedbe korekcij in zagotovitev integritete posredovanih dokumentov.
- Lažje obravnavanje in obvladovanje izjem, ki bo skrbnikom omogočale hitrejše odkrivanje vsebinskih napak pri sestavi dokumentov in povezovanju z šifrantov z zalednega sistema.

## 1.3 Uporabljene kratice

**COLD** COLD (ang. »Computer output to laser disk«) ali ERM (ang. »enterprise report management«) so sistemi namenjeni zajemu, arhiviranju, hranjenju in prevzetju večje količine podatkov, kot so računovodska poročila, evidence posojil, popis inventarja ipd... . Ti sistemi se izvajajo kot zamenjava za izdelavo fizičnih poročil in mikrofilmskega arhiviranja. [10]

<b>Microform</b>	Ureditev slik v pomanjšani obliki shranjenih na mikrofilmski trak, microfiche kartice ali mikrokartice. Zapis prvih dveh je urejen na fotografskem filmu, tretji pa na kartonastih karticah(ni več v uporabi)
<b>GZS</b>	Gospodarska zbornica Slovenije
<b>ZBS</b>	Združenje bank Slovenije
<b>DMS</b>	Dokumentarni informacijski sistem za hrambo in ureditev dokumentov z zmožnostjo sledenja spremembam in zagotavljanju integritete (ang. »Document management system«). [1]
<b>BPM</b>	Upravljanje poslovnih delovnih tokov (ang. »Business process management«)
<b>ERP</b>	Informacijski sistem za upravljanje informacij znotraj celotne organizacije. (ang. »Enterprise resource planning«)
<b>IS</b>	Informacijski sistemi
<b>OCR</b>	Strojna razpoznavna strojnih znakov z digitaliziranega dokumenta (ang. »Optical character recognition«).
<b>ICR</b>	Strojna razpoznavna ročne pisave z digitaliziranega dokumenta (ang. »Intelligent character recognition«).
<b>ZVDAGA</b>	Zakon o varstvu dokumentarnega in arhivskega gradiva ter arhivih
<b>UVDAG</b>	Uredba o varstvu dokumentarnega in arhivskega gradiva
<b>ETZ</b>	Enotne tehnološke zahteve
<b>MOREQ</b>	Zbirka zahtev za delo z elektronskimi zapisi (ang. »Model Requirements for the Management of Electronic Records«)
<b>ZEPEP</b>	Zakon o elektronskem poslovanju in elektronskem podpisu
<b>AIIM</b>	Neprofitna organizacija IT profesionalcev. (ang. »Association for Information and Image Management«)
<b>Tiff</b>	Format slikovne datoteke. (ang. »Tagged Image File Format«)
<b>PDF</b>	Format elektronskih dokumentov elektronskega ali digitaliziranega izvora. (ang. »Portable Document Format«)
<b>XML</b>	Format strukturirane elektronske datoteke. Struktura je določena za posamični IS ali pa je povzeta iz dogovorjenih standardov za prenos podatkov. (ang. »Extensible Markup Language«)
<b>XSL</b>	Strukturirana datoteka za določevanje prikaza XML dokumentov (ang.

	»Extensible Stylesheet Language«)
<b>CSV</b>	Tekstovna datoteka namenjena zapisovanju preglednic (ang. »Comma separated value«)
<b>PGP</b>	Odprt kriptografski standard za enkripcijo, dekripcijo, podpisovanje in urejanje ključev. [6]
<b>C# .NET</b>	Microsoftov programski jezik z uporabo .NET programskega ogrodja.

## **2 Zakonske zahteve in zahteve standardov**

Pri analizi in načrtovanju rešitve za zajem in pretvorbo dokumentov smo morali upoštevati zakonske omejitve in priporočila, ki nam jih narekuje zakonodaja Republike Slovenije ter priporočila standardov s področja zajema in pretvorbe dokumentov ter formatov, primernih za dolgoročno hrambo.

### **2.1 Zakonodaja s področja zajema in pretvorbe dokumentarnega gradiva**

Pri pripravi rešitve smo upoštevali določene zakonske zahteve, ki jih lahko opredelimo kot omejitve pri pripravi rešitve, saj le s popolno izpolnitvijo opredeljenih zahtev lahko zajamemo in pretvorimo dokument, ki bo pravno veljaven.

Rešitev mora zagotavljati zajem in pretvorbo dokumentov na način, ki bo zagotavljal temeljna načela zajema in pretvorbe gradiva, ki so napisana v zapisana v zakonu o varstvu dokumentarnega in arhivskega gradiva ter arhivih. [12]

- Načelo ohranjanja dokumentarnega gradiva oziroma uporabnosti njegove vsebine.
- Načelo trajnosti.
- Načelo dostopnosti.
- Načelo varstva kulturnega spomenika.

Poleg osnovnih načel, ki so opredeljene v ZVDAGA, pa moramo slediti še naslednjim zakonom in priporočilom:

- Uredba o varstvu dokumentarnega in arhivskega gradiva [13]

Ta uredba ureja delovanje in notranja pravila oseb, ki hranijo dokumentarno oziroma arhivsko gradivo, hrambo tega gradiva v fizični in digitalni obliki, splošne pogoje, registracijo in akreditacijo opreme in storitev za digitalno hrambo, odbiranje in izročanje arhivskega gradiva javnim arhivom, strokovno obdelavo in vodenje evidenc arhivskega gradiva, varstvo filmskega in zasebnega arhivskega gradiva, uporabo arhivskega gradiva v arhivih ter delo arhivske komisije.

- Enotne tehnološke zahteve. [2]

Podrobneje opredeljujejo poslovne, organizacijske in tehnološke pogoje za izpolnjevanje tega zakona in na njegovi podlagi izdanih podzakonskih predpisov.

Zbirka zahtev za delo z elektronskimi zapisi. [7]

Celovita specifikacija funkcionalnih zahtev, standardov in priporočil za delo z elektronskimi zapisi. Posebej specificira kako naj bi se med seboj povezovale klasifikacije datotek, zapisov, dokumentov, časa retenzije ipd. Primerna je za obe vrsti dokumentov, elektronskih in hibridnih (dokument v fizični in digitalni obliki).

## **2.2 Standardi s področja zajema in pretvorbe dokumentarnega gradiva**

Glavna omejitev pri pripravi rešitve za zajem elektronskih dokumentov je bila zagotovitev pretvorbe v obliko zapisa, ki bo ustrezala dolgoročni hrambi gradiva v digitalni obliki. Za zadostitev tej zahtevi smo morali rešitev oblikovati tako, da podpira zajem in pretvorbo formatov, ki so določeni kot veljavni formati za dolgoročno hrambo gradiva v digitalni obliki. Dolgoročna hramba gradiva pomeni hrambo gradiva daljšo od 5 let. Za to vrsto hrambo velja uporaba posebnih formatov zapisa, ki zagotavljajo, da bodo ti dokumenti dostopni in berljivi ter bo omogočena naknadna pretvorba v nove formate tudi po preteku več let ali desetletij.

Zakonodaja in priporočila za področje zajema in pretvorbe gradiva so jasno opredeljena, zaradi česa mora rešitev zagotavljati pravilno pretvorbo in možnost preverjanja pravilnosti pretvorbe za vsak tip dokumenta, ki je bil zajet s to rešitvijo (validacija formata zapisa). Rešitev pa mora zagotavljati možnost, da zajemamo dokument v obliki, ki je že primerna za dolgoročno hrambo. V tem primeru ne smemo posegati v samo strukturo dokumenta.

V nadaljevanju bomo opisali zahteve in primerne formate za dolgoročno hrambo. Predstavili pa bomo tudi nekatere tipe datotek oziroma formate zapisa, ki so najbolj pogosto predmet zajema in obdelave z našo rešitvijo.

### **2.2.1 Splošne zahteve oblike zapisa za dolgoročno hrambo**

Za obliko zapisa za dolgoročno hrambo se šteje taka oblika, ki izpolnjuje naslednje pogoje:

1. zagotavlja ohranitev vsebine gradiva tako, da pomeni urejeno celoto vseh potrebnih podatkov in povezav med njimi,
2. je široko priznana in uveljavljena oziroma uporabljana ter njena uporaba podprta z na trgu uveljavljeno strojno in programsko opremo,
3. neposredno uporabna za reprodukcijo vsebine ali enostavno pretvorjena v obliko, ki je neposredno uporabna,
4. omogoča samodejno pretvorbo iz najpogostejše uporabljenih izvornih oblik zapisa s samodejno zaznavo in poročanjem o nepredvidenih dogodkih oziroma napakah pri pretvorbi,
5. je neodvisna od posamezne programske ali strojne opreme oziroma okolja,
6. glede na stanje stroke obstaja velika stopnja verjetnosti, da zagotavlja varno hrambo več kot pet let,
7. omogoča po današnjih strokovnih predvidevanjih po tem obdobju pretvorbo v novo, takrat določeno obliko zapisa za dolgoročno hrambo,
8. temelji na mednarodnem, državnem ali splošno priznanem in praviloma odprtem standardu, če obstaja,
9. izpolnjuje druge zahteve iz zakona in te uredbe. [12]

### **2.2.2 Najpogostejše uporabljene oblike zapisa**

Oblika oziroma format zapisa je standardiziran način kako se kodirajo informacije dokumenta v datoteko za hrambo na elektronskem mediju, ki jih znamo uporabljati, prikazovati, manipulirati ipd... . Poznamo številne oblike zapisa za vsak tip dokumenta, naj bodo to sistemske, slikovne, tekstovne, ...

Odločitve glede izbire oblike zapisa so odvisne od različnih meril in izhodišč, zato se po svetu celo razlikujejo. Posamezne oblike zapisa imajo dobre in slabe lastnosti, zato jih je potrebno pred izbiro, za konkreten namen hrambe, poznati ter uskladiti z lastnimi zahtevami in dolgoročnim tveganjem v zvezi s temi lastnostmi. Tipičen primer je uporaba formatov, ki omogočajo stiskanje in s tem prihranek prostora. Stiskanje je lahko izgubno ali brez izgube. Slednje je zaželena rešitev za dolgoročno hrambo, še posebno za arhivsko gradivo, ki bo predano v pristojne arhive.

Primer oblike zapisa brez stiskanja je za avdiogradivo BWF (Broadcast Wave File), za grafične dokumente pa TIFF.



V spodnji preglednici so navedene najpogostejše uporabljene oblike, ki ustrezajo splošnim zahtevam za dolgoročno hrambo opisanih v UVDAG in zahtevam opredeljenih v ETZ.

<b>Formati za hrambo vsebine dokumentov</b>	
<b>Naziv formata</b>	<b>Opis</b>
<ul style="list-style-type: none"> <li>▪ PDF/A (ISO 19005)</li> <li>▪ W3C XML (ISO 8879)</li> <li>▪ ODF (ISO/IEC 26300)</li> </ul>	Oblika zapisa za tekstovne in mešane dokumente
<ul style="list-style-type: none"> <li>▪ HMTL (ISO/IEC 15445)</li> <li>▪ WARC (ISO 28500)</li> </ul>	Oblika zapisa za spletne vsebine
<ul style="list-style-type: none"> <li>▪ TIFF (ISO 12639 ver. 6)</li> <li>▪ JPEG (ISO/IEC IS 10918-1)</li> <li>▪ PNG (ISO/IEC 15948:2004)</li> <li>▪ JPEG2000 (ISO/IEC 15444)</li> <li>▪ SVG 2D v 1.1 W3C</li> <li>▪ DWG 3D (de facto standard)</li> </ul>	Oblika zapisa za grafične dokumente
<ul style="list-style-type: none"> <li>▪ ANSI/SMPTE 268M (DPX)</li> <li>▪ Motion JPEG2000 (ISO/IEC 15444-3)</li> <li>▪ FLAC</li> </ul>	Oblika zapisa za avdio, video, film (brezizgubno stiskanje)
<ul style="list-style-type: none"> <li>▪ MPEG-2 Audio Layer III (ISO/IEC 13818-3-MP3)</li> <li>▪ MPEG-2 Audio AAC (ISO/IEC 13818-7)</li> <li>▪ MPEG-4 Audio AAC (ISO/IEC 14496-3)</li> <li>▪ MEPEG- AVC (ISO/IEC 14496)</li> </ul>	Oblika zapisa za avdio, video, film (stiskanje z izgubami)
<ul style="list-style-type: none"> <li>▪ CCIT group 4 – črno-beli dokumenti</li> <li>▪ LZW – barvni dokumenti</li> <li>▪ ZIP</li> </ul>	Kompresija

<ul style="list-style-type: none"> <li>▪ Latin-2 (ISO/IEC 8859-2)</li> <li>▪ UTF-8 (ISO/IEC 10646)</li> </ul>	Kodirne tabele za tekstovne in mešane dokumente
---	---

Tabela 1.1: Dovoljene oblike zapisa za dolgoročno hrambo gradiva

V naslednjih poglavjih bomo predstavili standarde PDF/A, W3C XML in TIFF, ki so v naši rešitvi, poleg tekstovnih datotek, najbolj pogosto uporabljeni formati zapisa. Osredotočili pa se bomo še na dovoljene standarde kompresije, ki so priporočljive za uporabo pri arhiviranju dokumentov, saj omogočajo stiskanje brez izgube.

### 2.2.3 PDF/A – Oblika zapisa

PDF (ang. »Portable File Document«) je format datoteke, ki ga je lansiralo podjetje Adobe Systems company leta 1993. Specifikacija formata je bila javno dostopna in je ostala kontrolirana s strani podjetja Adobe, dokler ni bila objavljena kot odprt standard s strani ISO organizacije kot ISO 32000-1:2008.

PDF format, ki je bil zasnovan z namenom, da bi lahko elektronske datoteke lažje in neodvisno prenašali med sistemi in raznimi integracijami. Je strukturiran format, ki za prikaze povzema celoten opis fiksne postavitve dokumenta vključno z tekstovnimi, grafičnimi in ostalimi objekti, ki jih potrebuje za prikaz vsebine.

Zakaj je PDF primeren za dolgoročno hrambo:

- Omogoča shranjevanje metapodatkov, ki so lahko uporabljeni za avtomatsko klasifikacijo (ang »type metadata«). Npr. naslov, avtor, predmet vsebine, ključne besede, ipd.
- PDF datoteka hrani strukturirane objekte, ki so lahko uporabljeni za hitro iskanje.
- Datoteke so majhne in kompaktne, zaradi česa so primerne za hitre prenose do odjemalcev.
- Vsebina PDF strani je neodvisna od operacijskega sistema in prikazovalnika. S tem se pokaže prednost reprodukciji dokumenta (tiskanje, prikaz, ipd).

Zaradi popularnosti formata in rednih sprememb v novih verzijah se je pojavila ideja o kreaciji novega standarda za arhiviranje na osnovi PDF formata. Inicijativo za kreacijo novega standarda so leta 2002 objavili organizacije AIIM, NPES in Upravni urad ameriškega sodišča. Novi standard opisan v ISO 19005-1 dodaja mehanizem za ohranjanje predstavitev

elektronskih dokumentov, na način, ki čez čas ohranja vizualni prikaz dokumenta, neodvisno od orodij za kreacijo, hrambo ali interpretacijo.

Ključni element spremembe je bila obnovljivost dokumenta, zaradi česa mora biti datoteka po PDF/A standardu v celoti samozadostna in mora vsebovati vse elemente za ohranitev prikaza. To vključuje vse informacije povezane z vidnim tekstom, pisavo, barvo, slikami ipd. Standard prepoveduje uporabo kakršnekoli posredne ali neposredne uporabe zunanjih informacij, kot so slike, zvočni ali video zapisi. [5,16]

Ravni skladnosti:

- PDF/A – 1
  - PDF/A – 1b  
Vsebuje vse elemente za ohranitev vizualnega prikaza dokumenta.
  - PDF/A – 1a  
Vsebuje vse zahteve 1b skladnosti in dodaja hierarhijo strukture, oznake PDF datoteke, Unicode znakovno mapiranje in specifikacijo uporabljenega jezika.
- PDF/A – 2  
Dodaja funkcionalnosti verzij PDF 1.5, 1.6 in 1.7, ter dovoljuje JPEG2000 kompresijo, podporo za prozornost na objektih in dodatne ravni, možnost vključevanja drugih PDF/A datotek in PAdES standard.
- PDF/A – 3  
Dovoljuje vključevanje arbitrarnih datotek (npr XML, CSV, CAD, DOC, XLS ipd).

[5,16]

#### **2.2.4 TIFF – Oblika zapisa**

TIFF (ang. »Tagged Image File Format«) format datoteke, ki ga je lansiralo podjetje Aldus Corporation leta 1986 z namenom, da bi ustvarili standard za shranjevanje digitaliziranih dokumentov. Format zapisa in specifikacija je sedaj kontrolirana s strani Adobe Corporation, vendar ni bilo nobene večje spremembe od leta 1992. Format je pripravljen tako, da bi ga lahko za potrebe novih tehnologij zelo hitro nadgradili in zadostili potrebam arhiva. Microsoft

je predstavil tudi možnost razširitve z dodajanjem tekstovne iskalne vsebine (ang. »Text searchable TIFF«), vendar ta razširitev ni dobro podprta pri ostalih proizvajalcih.

Pomembno je razumeti, da je TIFF datotečni format zapisa in ne slikovni format. Predstavljamo si ga lahko kot odlagališče za eno ali več slik, ki so lahko različnega tipa. V spodnji tabeli lahko vidimo najbolj pogoste sheme slik, ki so lahko vključeni v TIFF datoteki.

<b>Sheme uporabljene v TIFF obliki zapisa</b>	
<b>Shema kompresije</b>	<b>Tipična raba</b>
Group4	Najbolj pogosto uporabljena za dokumente digitalizirane v črno-bele tehniki.
Group3	Uporabljen za pošiljanje faksov.
JPEG	Uporabljeno za digitalizacijo barvnih in sivinskih dokumentov.
LZW	Uporabljeno za digitalizacijo barvnih in sivinskih dokumentov.
Brez stiskanja (ang: »Uncompressed«)	Najbolj pogosto uporabljen kot izhod grafičnih programov.
Ostalo	Ostale manj pogosto uporabljene sheme vključujejo ZIP, Packbits in RLE algoritme.

Tabela 2.2.3.1 Uporabljene sheme v TIFF obliki zapisa

V TIFF datoteki so lahko shranjeni tudi metapodatki z uporabo polj za oznake (ang. »Tag Fields«). Obstaja niz osnovnih polj, ki jih morajo podpirati vse aplikacije za uporabo TIFF formata. Osnovni niz polj označuje strojno opremo kreacije, čas programske opreme ipd. Obstaja tudi razširitveni niz, podprta pa so tudi »privatna« polja, ki so namenjena za implementacijo v programskih oprelih.

TIFF je bil med prvimi formati, ki je bil primeren za dolgoročno hrambo, sedaj pa se pri izbiri formata med TIFF in PDF/A datoteko, odločamo bolj glede izvora dokumenta (digitaliziran ali elektronski dokument), ter potreb in zahtev zalednih ali arhivskih sistemov (velikost dokumenta, iskalna vsebina, zmožnosti prikaza, uporaba v zalednem sistemu ipd...) [4,5]

### 2.2.5 WC3 XML – Oblika zapisa

XML (ang. »eXtensible Markup Language«) dokument je tekstovna datoteka, ki je zapisana v označevalnem jeziku (ang »markup language«), zelo podobno kot pri HTML standardu. [17]

XML format je bil ustvarjen z namenom prenosa in hrambe podatkov, ne pa prikazu le teh. Za boljši uporabniški prikaz se uporablja XLST vizualizacijska datoteka, s katero lahko dokument pretvorimo v HTML standard, ki je namenjen prikazu podatkov.

XML format sestavlja drevesno strukturo, ki se začne s korensko oznako in se veji do končnih oznak »listov«. Nima pa določenih nobenih standardnih imen za oznake, te gradi razvijalec, za katere posreduje definicije vsem, ki bodo uporabili dokument za prenos podatkov.

Primer preprostega XML dokumenta:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Document>
    <Ime>Marko</Ime>
    <Priimek>Žankar</Priimek>
</Document>
```

»Document« označuje korensko oznako (opisno bi pomenilo, da je vsebina XML datoteke dokument).

»Ime« in »Priimek« tu označujeta podrejene elemente, ki predstavljajo oznake za vsebino podatkov.

XML dokument je eden izmed najbolj pogosto uporabljenih formatov v večini programskih rešitev, vendar pa ne pomeni, da je vsebina vseh XML datotek pomembna za nadaljnjo hrambo. Med ključne uporabe lahko štejemo SOAP sporočila spletnih servisov, dokumentno pa je v Sloveniji med bolj ključnimi dokumenti tako urejen format za prenos elektronski računov. [17]

### 2.2.6 Dovoljeni standardi kompresije

V tem poglavju bomo kratko predstavili kompresije, ki so zapisane v ETZ. Vse tri kompresije uporabljajo algoritme za breizgubno stiskanje, zaradi česa so bolj primerne za uporabo pri dolgoročni hrambi gradiva, saj lahko format zapisa, ki ga stiskamo povrnemo v prvotno stanje.

#### **2.2.6.1 CCITT GROUP 4**

CCITT Group 4 kompresija je uporabljena za samo črno bele dokumente. Nudi izboljšavo algoritma CITT Group 3 z odstranitvijo znakov za konec vrstice (ang. »EOL Code«).

Algoritem deluje z zapisom različno dolgih kodnih besed. Te kodne besede v vrstici predstavljajo dolžino črnih ali belih točk v sliki (dolžino lahko predstavlja tudi več kodnih besed). Zapisi kodnih besed so narejeni tako, da se označevanje dolžin za barve izmenjuje. Na začetku vsake vrstice pa je vedno na prvem mestu opisana dolžina za belo barvo, da pri dekompresiji ali prikazu ne pride do težav z barvno sinhronizacijo. V primeru, da se vrstica začne z črno točko se najprej vpiše ničelno dolžino za zapis bele barve. [4]

#### **2.2.6.2 LZW**

LZW (ang. »Lempel–Ziv–Welch« compression algorithm) je tip kompresije, ki ga uporabljamo za sivinsko in barvno digitalizirane dokumente. Objavil ga je Terry Welch leta 1984, kot izboljšano verzijo LZ78 algoritma.

Algoritem temelji na prevajalni tabeli, ki vhodni niz znakov preslika v kodo. Implementacija, ki jo uporablja oblika zapisa TIFF uporablja spremenljivo dolžino kode, vendar ne več kot 12 bitov. Prevajalna tabela je različna za vsak trak vhodnih nizov, vendar je ne potrebujemo hraniti, ker lahko pri dekompresiji sestavimo enako tabelo.

Karakteristike kompresije:

- Deluje na rasterskih slikah z različnimi barvnimi globinami.
- Lahko upravlja s široko paleto ponavljajočih vzorcev.
- Je razmeroma hiter tako za stiskanje kot za razširitev.
- Ima razumno obnašanje v najslabših primerih

Ne potrebuje strojne ali programske opreme za računanje z plavajočo vejico.[4]

### 2.2.6.3 ZIP

ZIP je eden najbolj razširjenih stisnjenih oblik zapisa. Vsesplošno je uporabljena za združevanje, stiskanje in šifriranje datotek v en interoperabilen zabojnik.

Vsaka datoteka znotraj ZIP zaboja je lahko posamično stisnjena, shranjena (nestisnjena datoteka), kriptirana ali digitalno podpisana, ne glede na ostale podatkovne datoteke s katerimi si deli zabojnik. Vsaka datoteka lahko vsebuje različne metode za brezgubno stiskanje, ki se zapišejo v glavo vsake datoteke.

Struktura formata:

- Zapis datoteke (eden ali več zapisov)
  - Glava datoteke
- Podatki o vključeni datoteki (ime, čas in datum spremembe, CRC32 za zagotavljanje celovitosti, min verzija za uporabo, velikost, metoda kompresije ipd.)
  - Glava enkripcije
  - Podatki datoteke
  - Opisni podatki
- Glava dekripcija arhiva
- Dodatni zapisi arhiva
- Glava centralnega direktorija (eden ali več zapisov)
- ZIP64 končni zapis centralnega direktorija
- ZIP64 končni zapis centralnega lokatorja
- Končni zapis centralnega direktorija [11]

## 2.3 Zahteva za metapodatkovno shemo

### 2.3.1 Opredelitev in namen

Metapodatek je podatek, ki vsebuje informacijo o drugih podatkih. Pogosto se uporabljajo za iskanje ali upravljanje informacijskih virov z abstrakcijo, klasifikacijo ali z zajemom informacije, ki ni vključena v viru. Njegov glavni namen je povezovanje zapisov z dokumenti v zalednem sistemu. Uporabljajo pa se tudi kot izmenjave določenih podatkov, zapisov dogodkov ali pa tehničnih lastnosti dokumenta. [15]

Običajno se metapodatke klasificira v določene kategorije, ki se opirajo na dogovore o vrednosti le teh.

Tipično jih delimo na štiri kategorije kategorije:

- Administrativni ali upravni metapodatki  
Vključujejo datum in izvor zajema, datum in metodo uničenja.
- Opisni metapodatki  
Vključujejo podatke o vsebini zapisa.
- Ohranitveni metapodatki  
Lahko shranjujejo vse zapise aktivnosti, ki zaščitijo ali podaljšajo obstojnost zapisov.
- Strukturni metapodatki  
Podatki, ki lahko označujejo medsebojne povezanosti zajetih informacijskih virov, kot so številke strani pri večstranskih dokumentih ipd... .

Glavni namen metapodatkov je hraniti informacije o zapisih za potrebe iskanja in povezovanje dokumentov oziroma izmenjavo podatkov v zalednih sistemih.

### 2.3.2 Minimalne zahteve

Gradivo, pripravljeno za zajem, mora biti pred pretvorbo evidentirano oz. opremljeno z metapodatki. Metapodatki se lahko prenašajo z obstoječih evidenc o gradivu, črtnih kod, z uporabo OCR programov, se vnesejo ročno ali pripnejo samodejno.

Organizacija mora z Notranjimi pravili (interni pravni akt poslovnega subjekta, ki predpisuje način in hrambo gradiva poslovnega subjekta v elektronski obliki) opredeliti seznam obveznih metapodatkov za vsako vrsto/tip gradiva in način njihovega vnosa (samodejno, ročno, uparjanje, OCR zajem ipd.). Ponudnik opreme in storitev (ponudnik storitev zajema in e-hrambe gradiva ter spremljevalnih storitev) je dolžan določiti minimalen nabor metapodatkov, ki jih bo zajemal oz. hranil za določen poslovni subjekt. Nabor metapodatkov se lahko s pogodbo o izvajanju storitve tudi razširi. [2]

Zajem in specifikacija metapodatkov za vsak tip dokumentov je dogovorjena z naročnikom storitve. V tej specifikaciji se skupaj z naročnikom opredeli nabor, ki ga bomo zajemali ter pravila zanj. Določujemo jih predvsem glede na potrebe zalednih sistemov, ki bodo dokumente lahko povezovali z zapisi v sistemu ali pa bodo uporabniku olajšale iskanje in jim s tem pohitrili delo.

V spodnji tabeli je predstavljen izveček minimalnih zahtev za metapodatke za različne oblike zapisa.

Besedilni in mešani dokument	Zvočno gradivo
------------------------------	----------------



<ul style="list-style-type: none"> <li>▪ enolična identifikacijska oznaka</li> <li>▪ naslov ali kratka oznaka vsebine</li> <li>▪ datum (prejetja, nastanka)</li> <li>▪ avtor oz. pošiljatelj</li> <li>▪ naslovník (prejemnik)</li> </ul>	<ul style="list-style-type: none"> <li>▪ enolična identifikacijska oznaka</li> <li>▪ kraj nastanka</li> <li>▪ naslov</li> <li>▪ vsebina</li> <li>▪ čas nastanka</li> <li>▪ ustvarjalec</li> <li>▪ izvorni format</li> <li>▪ izvorna dolžina</li> <li>▪ vrsta nosilca</li> </ul>
<b>Film in avdiovizualno gradivo</b>	<b>Spletne strani</b>
<ul style="list-style-type: none"> <li>▪ enolična identifikacijska oznaka</li> <li>▪ naslov</li> <li>▪ leto nastanka,</li> <li>▪ zasedba (igralska in celotne ekipe)</li> <li>▪ produkcija</li> <li>▪ producent</li> <li>▪ država nastanka filma</li> <li>▪ izvorni format</li> <li>▪ izvorni nosilec</li> <li>▪ izvorna dolžina</li> <li>▪ izvorni jezik</li> <li>▪ serija/serija</li> <li>▪ žanr</li> <li>▪ zahvale za delo pri filmu</li> <li>▪ povezave</li> <li>▪ vir</li> </ul>	<ul style="list-style-type: none"> <li>▪ izvor</li> <li>▪ mesto objave</li> <li>▪ obliko</li> <li>▪ pogostost</li> <li>▪ postopek zajema in e-hrambe</li> <li>▪ obseg potrebnih metapodatkov</li> <li>▪ seznam funkcionalnosti, ki se ob zajemu in e-hrambi ohranjajo</li> <li>▪ odgovorne osebe</li> <li>▪ enolična identifikacijska oznaka (evidenčna oznaka)</li> <li>▪ predmet (zadeva) <ul style="list-style-type: none"> <li>▪ sestava</li> <li>▪ kontekst dokumenta in mesto objave</li> <li>▪ identifikator dokumenta</li> </ul> </li> </ul>
<b>Elektronska pošta</b>	
<ul style="list-style-type: none"> <li>▪ enolično identifikacijsko oznako (evidenčna oznaka)</li> <li>▪ izvor (e-poštni predal)</li> <li>▪ obliko zapisa e-hrambe</li> <li>▪ način zajema</li> <li>▪ obseg potrebnih metapodatkov</li> <li>▪ odgovorne osebe</li> <li>▪ naslov poštnega predala prejemnika odgovorov (polje »From« v glavi sporočila)</li> <li>▪ naslov poštnega predala pošiljatelja e-sporočila (polje »Sender« v glavi sporočila)</li> <li>▪ naslov prejemnika e-sporočila (polje »To« v glavi sporočila)</li> </ul>	

- |   |
|---|
| <ul style="list-style-type: none"><li>▪ naslov/predmet e-sporočila (polje »Subject« v glavi sporočila)</li><li>▪ datum e-sporočila (polje »Date« v glavi sporočila)</li><li>▪ identifikator e-sporočila (polje »Message-ID« v glavi sporočila)</li><li>▪ število priponk</li><li>▪ za vsako priponko identifikator osnovnega elektronskega sporočila</li><li>▪ zastavica, da je bil dokument spremenjen</li><li>▪ informacija o prikrivalnem postopku (enkripciji)</li><li>▪ informacija o elektronskem podpisu</li><li>▪ poštni predal, iz katerega je bil narejen zajem</li></ul> |
|---|

Tabela 2.3.2.1: Minimalne zahteve za metapodatke za različne oblike zapisa

# 3 Razvoj programske opreme

## 3.1 Predstavitev metodologije - Sashimi Waterfall Model

### 3.1.1 Waterfall model

Waterfall programsko razvojna metodologija je izmed najbolj poznanih in uporabljenih metodologij. Prvotno je bila zasnovana za predelovalno in gradbeno industrijo in je dobila ime »Waterfall« (slap) zaradi načina prehoda med fazami, ki deluje kot pretok slapa. Vse faze metodologije si sledijo po vrstnem redu, pričetek nove faze pa lahko naredimo šele, ko se prejšnja zaključi (projekti mejnik) in s tem ne dopušča vračanja. Najbolj je uporabna v projektih, kjer so zahteve jasno navedene in se ne spreminjajo, ter imamo dogovorjeno časovnico in proračun razvoja. [14]

Večja težava, ki se pojavlja s tradicionalno waterfall metodologijo, je nemogoča popolno dokončanje ene same faze preden opravimo prehod na novo. Druga večja težava je narava načrtovanja in kreacije na vseh področjih, vključno z programskim razvojem, ki zelo otežuje in onemogoča definicijo vseh zahtev vnaprej in pred končanjem projekta. V kolikor se zahteve projekta vedno spreminjajo ali pa se dodajajo nove ideje potem se model waterfall-a ne ujema.

### 3.1.2 Sashimi Model

Sashimi (ime je pridobil po japonski jedi, kjer se servirani kosi prekrivajo) model je objavil Peter DeGrace, ter ga pogosto imenujejo waterfall model z prekrivajočimi fazami ali waterfall z povratno informacijo (ang "waterfall model with overlapping phases" in "the waterfall model with feedback"). Model je nastal zaradi večjih pomanjkljivosti v klasičnem modelu ter kritik na njegov račun.

Glavna sprememba v primerjavi s z klasičnim modelom je dodana možnost prekrivajočih se faz. Zaradi te spremembe je omogočena večja fleksibilnost, ki omogoča, da se funkcije projekta izvajajo sočasno. S to pridobitvijo hitreje odpravimo vse slabosti programske opreme že v razvojnem ciklu, ter s tem prihranimo dodatne stroške, ker smo odpravili pomanjkljivosti preden smo vstopili v fazo implementacije.

Zaradi možnosti delovanja faz sočasno so nam omogočene tudi spremembe pri osnovnem načrtu. Vseeno pa je potrebna pazljivost, da se ne vračamo več kot eno fazo nazaj. To bi

pomenilo, da vse ni bilo zajeto pri opredeljevanju zahtev projekta in se lahko odraža kot draga napaka pri fazi načrtovanja.

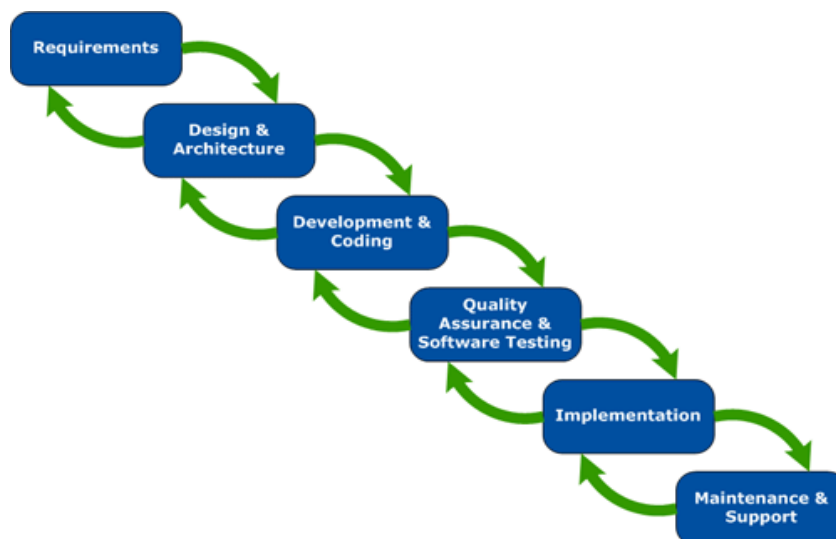
Druga prednosti modela je bolj sproščen pristop zaradi lažjih procedur, dokumentov in pregledov. To omogoča razvojni ekipi, da svoj čas posveča pri programiranju končnega produkta, ker se jim ni potrebno ukvarjati s posebnimi procedurami.

### 3.1.3 Faze Sashimi Waterfall Model modela

Sashimi Waterfall Model sestavlja več faz- Te faze so:

1. Analiza zahtev
2. Načrt in arhitektura
3. Razvoj in kodiranje
4. Testiranje in zagotavljanje kakovosti
5. Implementacija
6. Vzdrževanje in podpora

Na spodnji sliki so shematsko prikazane posamezne faze modela.



Slika 3.1.3.1: Prikaz waterfall metodologije po fazah (<http://www.managedmayhem.com>)

### **3.1.3.1 Analiza zahtev**

Prvi korak in eden izmed najbolj pomembnih opravil pri zasnovi programske opreme je zbiranje in določevanje zahtev za celoten projekt. Ob zaključku te faze bi morali imeti vsi sodelujoči (vključno z naročnikom) osnovno razumevanje, kakšno programsko opremo je potrebno razviti. Tu je naloga izvajalca tudi svetovanje, saj mora uskladiti nepopolne, dvoumne ali nasprotujoče si zahteve. V tej fazi se lahko zahteve zapišejo bolj opisno in ni potrebno, da so tehnične narave. Za manjše projekte lahko že tu zastavimo časovnico in projektne mejnike, pri srednjih ali večjih projektih pa je priporočljivo, da to opravimo v fazi načrtovanja, kjer bodo bolj podrobno znana vsa potreba dela na projektu. [14,18]

### **3.1.3.2 Načrt in arhitektura**

Predno lahko pričnemo s programiranjem rešitve se moramo osredotočiti na izbrane zahteve ter pripraviti načrt za razvoj. V tej fazi pripravimo funkcionalno in tehnično specifikacijo, časovnico, zastavimo ključne mejnike ter dodelimo sredstva projektu. Iz načrta oziroma specifikacije mora biti razvidno katera platforma bo uporabljena, kakšen bo delovni tok aplikacije, specifikacija strojne opreme ipd.

Za lažjo komunikacijo z naročnikom (sponzor projekta) in razvojno ekipo se tu lahko tudi že pripravi grafični vmesnik ali pa UML načrt (»Unified Modeling Language«), ki bo pripomogel k lažjem dialogu, ter pri osnovnem razumevanju kako bo rešitev delovala. [14,18]

### **3.1.3.3 Razvoj in kodiranje**

Po sestavi celotnega načrta se lahko lotimo procesa, ki je časovno najbolj obsežen. Razvijalec ali ekipa razvijalcev se v tej fazi loti programiranja programske opreme. Za izvedbo tega procesa je ključno, da sta bili prvi dve fazi opravljeni pravilno. V kolikor bi pri njih prišlo do napake, bi to lahko pomenilo zamik projekta ali pa celo ponovno programiranje na že opravljenih delih. Ker pa opravljamo delo z prekrivajočim fazam tu lažje odpravljamo napake, saj se načrtovanje in arhitektura zaključi šele, ko so bile opravljene vse njene zahteve. [14,18]

### **3.1.3.4 Testiranje in zagotavljanje kakovosti**

V tej fazi v klasičnem modelu opravljamo testiranje nad zaključenim izdelkom, ki je bil izdelan v prejšnji fazi. Zaradi uporabe tega modela v primerjavi s klasičnim, je omogočeno, da je testiranje že del razvoja, kot je lahko tudi del implementacije. To pomeni, da razvijamo

in testiramo dokler testiranje ne zaključi faze razvoja, enako bi bilo pri implementaciji, dokler nimamo aplikacije popolno testirane in nameščene.

Poznamo pa naslednje načine testiranja:

- Testi enote (ang »Unit test«)
- Integracijski testi
- Sistemski testi
- Regresijski testi
- Funkcijski testi(ang »Functional/Acceptance Tests«)
- Testiranje zmogljivosti
- Testiranje obremenitve
- Testiranje skladnosti [14,18]

### **3.1.3.5 Implementacija**

V tem delu postavljamo produkt na končni sistem oziroma pripravimo ustrezno distribucijo programske opreme. Tu je potrebno pregledati in dokončati vso projektno dokumentacijo vključno z uporabniškimi navodili. Po potrebi pa se v tej fazi opravlja še šolanje končnega uporabnika.

Proces implementacije je edini proces pri katerem se lahko zgodi, da se prekriva z dvema predhodnima fazama, ker obstaja možnost, da pri testiranju in implementaciji še odkrijemo napako, ki jo je potrebno odpraviti z razvojnim procesom. [14,18]

### **3.1.3.6 Vzdrževanje in podpora**

Proces vzdrževanja in podpore je pomembnejši od samega zaključka projekta. Tu je priporočljivo, da se tehnični in vsebinski del programske opreme preda v skupino za podporo, saj je zagotavljanje delovanja ključno.

V okviru te faze lahko čez čas zberemo želje po novih funkcionalnosti ali odkrijemo napake v robnih primerih. [14,18]

## **3.2 Razvoj**

### **3.2.1 Zasnova programske rešitve**

Do zasnove programske opreme nas je pripeljalo povpraševanje po zajemu za več kot 10 različnih tipov dokumentov elektronskega izvora. Pri običajni praksi bi to pomenilo, gradnjo enega ali več programskih servisov, ki bi opravljali specifično delo ter ne bi morali biti posredovani v ponovno uporabo za nova povpraševanja.

Po analizi in pregledu vseh zahtev smo se odločili pregledati možnost izdelave univerzalnega servisa, ki bi vključeval najbolj uporabljene funkcionalnosti, a bi ohranjal možnost lažjega dodajanje novih, kar bi omogočalo tudi izvajanje novih vrst obdelav.

Pri načrtovanju smo prišli do ugotovitve, da bi krovna ali strežniška aplikacija lahko vsebovala le osnovne funkcionalnosti in ohranila glavni tok poteka dela. Funkcionalnosti, potrebne za separacijo, validacijo(zajem metapodatkov), pretvorbo in izvoz, pa bi ločili v posamične manjše projekte, ki se lahko vključujejo v matično aplikacijo.

Na osnovi analize in načrtovanja smo se odločili, da rešitev ločimo na tri posamične dele.

- Krovna/Strežniška aplikacija
  - Omogoča osnovne funkcionalnosti za ravnanje z datotekami
  - Omogoča ravnanje z izjemami
  - Omogoča testiranje in pripravo obdelave
  - Tok delovanja in klici vtičnikov
- Vtičniki
  - Razdeljeni na 4 podtipe (Separacija, validacija, pretvorba in izvoz)
  - Igrajo ključno vlogo pri obdelavi saj prinašajo funkcionalnosti servisu
- Skupna knjižnica za aplikacijo in vtičnike
  - Vključuje API vmesnik za gradnjo in klice vtičnikov
  - Vključuje podatkovne strukture objektov potrebnih za prenos med vtičniki in krovno aplikacijo
  - Vključuje splošne funkcionalnosti, ki so uporabljene v krovni aplikaciji, za pomoč pri gradnji vtičnikov

### **3.2.2 Funkcionalna specifikacija**

#### **3.2.2.1 Zahteve vmesnika**

- Vmesnik mora omogočati testiranje in možnost zagona obdelave za izbrano datoteko.

- Vmesnik mora omogočati preprosto kreacijo in manipulacijo vseh obdelav znotraj instance.
- Omogočati mora preverjanje vseh navedenih poti v nastavitvah.
- Omogočati mora manipulacijo delavnega toka aplikacije. (Določitev in sprememba vrstnega reda vtičnikov).
- Omogočati mora klic nastavitvenega vmesnika za vsak posamezen vtičnik, ter hraniti njegove nastavitve znotraj krovne obdelave.
- Omogočati mora filtre in razpoznavo DLL knjižnice, ki določujejo vtičnike.
- Pripravljen mora biti modul za testiranje, kjer bo omogočeno testiranje posamičnih in združenih vtičnikov.
- Omogočati mora nastavitev revizijske sledi z zapisi v bazo obdelave, ter kreacijo posebnih datotek, ki jih predložimo dokumentu.

### 3.2.2.2 Delovanje

Krovna aplikacija mora omogočati delovanje v treh ločenih načinih.

- Uporaba oziroma pogon obdelave čez uporabniški vmesnik.
  - Možnost izbire obdelave.
  - Določevanje ene ali več datotek za obdelavo.
  - Kratko poročilo o obdelavi (čas obdelave, število datotek, ipd),
- Servisno delovanje

Delovanje v načinu windows servisnega modula, ki avtomatsko prebira nastavljene poti za nove dokumente, ter ob najdenih datotekah proži obdelavo.

- Določljiva časovna perioda ob katerih bo narejen vpogled za nove datoteke.
  - Obdelava mora imeti možnost izklopa znotraj servisnega delovanja
  - Ravnanje z napakami ločuje datoteke z napako, ter obvesti odgovorno osebo
- Konzolna aplikacija (za lažjo uporabo v MS scheduled tasks)
  - Omogoča klice obdelave znotraj ukazne vrstice.
  - Omogoča klice vseh omogočenih obdelav.
  - Posredovanje poročila ob končanju



### 3.2.2.3 Splošna funkcionalnost

Ravnanje z datotekami:

- Aplikacija mora omogočati shranjevanje varnostne kopije vseh prejetih datotek, s pomočjo katere bo lahko administrator aplikacije preveril ob vsebinsko prijavljenih napakah, kaj je razlog napake.
- Aplikacija mora za varnostne kopije imeti nastavljen čas po kateri se bodo datoteke avtomatsko brisale (npr. 30 dni).
- Vsaka obdelava mora zagotavljati svojo delavno mapo, ki mora imeti omogočeno delovanje samo na lokalnem računalniku ali strežniku, kjer je aplikacija postavljena (zagotavljanje hitrega delovanja obdelav).
- Ravnanje z izjemami mora omogočati ločitev dokumentov v mapo za napake.
- Aplikacija mora za vhodno datoteko omogočati klic sekundarne obdelave (sekundarno obdelavo uporabimo, če želimo hraniti datoteke pred separacijo).

Revizija in obveščanje:

- Vse obdelave morajo omogočati nastavitve za obveščanje o poteku obdelave po elektronskih poštah, ki jih prejema vsebinsko odgovorne osebe.
- Obvestila o uspešnosti morajo zagotavljati strukturo, po kateri bo uporabnik lahko hitro razbral vrsto in datoteko obdelave, pridruženo z kratko statistiko obdelave (število dokumentov, vzrok napake ipd...).
- Obdelava mora omogočati vodenje revizije in statistike dokumentov z zapisi v podatkovni bazi.
- Aplikacije mora biti nastavljiva za različno strukturo statistične baze.
- Obdelava mora za vsak dokument zagotoviti revizijsko datoteko v XML strukturi, ki mora shranjevati vse podatke o dokumentu:
  - Vrednosti metapodatkov.
  - Dogodki obdelave z točno časovno oznako.
  - Izvlečki datotek pred in po pretvorbi vsaki vključeni pretvorbi (SHA).
  - Izvleček vhodne datoteke pred separacijo.
  - Ime vhodne datoteke.
  - Podatki o paketu iz katerega izhaja dokument (število dokumentov, velikost, ipd...).

Metapodatki in kontrola:

- Vsebina metapodatka mora imeti možnost nastavljive privzete vrednosti:
  - Poljuben tekst

- Ime datoteke
  - Datum obdelave
  - Čas obdelave
  - GUID (unikatna vrednost)
- Pred izvoznim delom mora aplikacija omogočati kontrolo vsebine metapodatkov z uporabo regularnih izrazov.
  - Vrednost metapodatkov je lahko uporabljena kot vhodni parameter sekundarne obdelave dokumentov (obdelava izvorne datoteke).

### 3.2.2.4 Vtičniki

- Razvijalcem je potrebno omogočiti razvoj zunaj projekta krovne aplikacije z vključitvijo skupne knjižnice.
- Grajene morajo biti z vključitvijo API vmesnika s katerim se določuje tip in način delovanja vtičnika.
- Razvijalci se morajo držati skupnih načel gradnje vtičnika ter ne smejo zapirati napak, temveč jih sporočati krovni aplikaciji.
- Hraniti morajo strukturo objekta oziroma razreda za interne nastavitve vtičnika.
- Funkcionalnosti in omejitve vtičnika morajo biti ustrezno dokumentirane.
- Omejiti mora dovoljene tipe dokumentov, ki jih sprejema v vtičnik.
- Funkcionalnost vtičnika mora ločena glede na tip vtičnika zaradi možnosti ponovne uporabe kot gradnik druge obdelave:
  - Vtičnik za uvoz in separacijo dokumentov.
- Prezem datotek z različnih sistemov, dekompresije stisnjene datoteke, dekripcija prevzete datoteke in ločevanje datoteke v logične dokumente.
  - Vtičnik za zajem in validacijo podatkov
- Zajeti mora podatke na dogovorjen način z datoteke ali z uporabo predhodnih vrednosti obstoječih metapodatkov.
  - Vtičnik za pretvorbo
- Omogočati mora dogovorjeno pretvorbo datotek dokumenta.
  - Vtičnik za izvoz
- Izvoz v dogovorjene zaledne sisteme.

### 3.2.2.5 Skupna knjižnica za aplikacijo in vtičnike

- Vključevati mora metode, ki bodo integratorju v pomoč gradnji novega vtičnika:
  - Metode za serializacijo in deserializacijo objektov (shranjevanje nastavitev v XML datoteko)
  - Metode za vodenje datoteke dogodkov (ang. »Log file«), ki se sovpada z datoteko krovne aplikacije.
  - Metode za vodenje natančnih dogodkov poteka, z namenom odkrivanja napak (ang. »debug file«)
  - Splošne metode, ki bi lahko olajšale delo
- Verzija gradnje oziroma nadgradnje knjižnice mora biti vodena ločeno od krovne aplikacije. Z omenjenim lahko zagotavljamo delovanje starih vtičnikov v kolikor ni prišlo do spremembe na vmesniku.
- Hrani celoten API vmesnik, ki je uporabljen znotraj krovne aplikacije in vseh pripadajočih vtičnikov, s katerim zagotavljamo dogovor in pravilno komunikacijo med vtičniki in krovno aplikacijo.
- API vmesnik mora zagotoviti kreacijo naslednjih metod in lastnosti vtičnika:
  - Naziv
  - Opombe oziroma kratek vpis vtičnika
  - Avtor
  - Datum kreacije oziroma zadnje spremembe
  - Verzijo
  - Metodo za klic in shranjevanje nastavitev vtičnika (splošna za vse vrste vtičnikov)
  - Metodo ali metode potrebno za posamičen korak obdelave (različno za tip vtičnika)
- Hrani celotno podatkovno strukturo za dokumente oziroma seznam dokumentov, ki bo uporabljena za prenos podatkov in informacij med posamičnimi vtičniki in krovno aplikacijo. Podatkovna struktura mora vključevati naslednje:
  - Zbirko datotek dokumenta, ter njihove polne poti znotraj začasne mape.
  - Zbirko metapodatkovnih polj z nazivi in vrednostjo metapodatka zajetega z dokumenta.
  - Metode za iskanje znotraj zbirke metapodatkov za lažjo uporabo znotraj vtičnikov.
  - Zgodovino dogodkov, ki so bili opravljeni nad datotekami dokumenta za potrebe vključevanja v revizijsko sled.

- Polje za opombo, ki jo lahko nastavi vtičnik za potrebo vključitve v revizijsko sled.

### **3.2.3 Opis najbolj pogosto uporabljenih vtičnikov**

#### **3.2.3.1 Vtičniki za uvoz in separacijo dokumentov**

##### *TEXT SEPARATOR*

Vtičnik za separacijo tekstovnih dokumentov. Separacijo omogoča na dva načina:

- Po številu vrstic.  
Administrator aplikacije določi vrednost vrstic, ki bo uporabljena za ločen dokument (1 – N).
- Glede na ločitveni znak ali niz znakov.  
Vtičnik po dokumentu išče nov niz, ki je bil vnesen v nastavitve. Ko je bil iskan niz najden zaključi trenutni dokument ter prične z pisanjem novega.

Dodatno omogoča združevanje dokumentov, če ima vklopljeno možnost za primerjavo vrstic ali zaporedne besede v vrstici (npr.; Dvostranski račun, pri katerem na enakem mestu iščemo številko računa).

##### *XML SEPARATOR*

Vtičnik za separacijo XML datotek. Dokumente ločuje glede na objekte v izbranem vozlišču. V začasno mapo si najprej ustvari predlogo brez objektov v vozlišču (ostale vrednosti v dokumentu ostanejo enake), nato pa za vsak dokument uporabi svoj objekt in ga zapiše v predlogo.

Za izbiro vozlišča se uporablja XPATH poizvedbeni jezik (ang. »XML PATH LANGUAGE«), ki omogoča izbiro posamičnih vozlišč ali vrednosti znotraj XML datoteke. Tovrstna tehnologija je bila uporabljena zaradi možnosti nastavljivega iskanja, brez dodatne integracije.

##### *EINVOICE SPEARATOR*

Vtičnik za separacijo in uvoz elektronskih računov z ovojnico. V aplikacijo sprejema vse XML datoteke, ki jih ločuje na ovojnice in priloge. Po identifikaciji ovojnice pregleda njeno vsebino ter iz nje kreira ciljne dokumente (zadnja verzija ovojnice lahko vsebuje več

elektronskih računov). Pred posredovanjem v naslednji modul dodeljuje dokumentom še njihove priloge (zapisane v ovojnici za vsak račun).

Identifikacija ovojnice in nastavitve za priloge se zapisujejo kot nastavitve modula, ki se berejo z posredovane XML datoteke z uporabo XPATH poizvedbenega jezika.

#### *ZIP IMPORTER*

Vtičnik za uvoz dokumentov znotraj XML datoteke. Za ločene dokumente uporabi vse datoteke znotraj stisnjene datoteke.

Dodatni možnosti:

- Združevanje datotek z enakim imenom (različne končnice) v skupni dokument.
- Določitev primarne končnice ob združevanju dokumentov.

#### *PGP DECRYPTOR*

Vtičnik za uvoz PGP kriptiranih datotek. Za dekriptacijo je uporabljen PGP standard [6] (ang »Pretty Good Privacy«). V nastavitvah sprejema lokacijo privatnega ključa ter geslo za dekriptacijo datotek.

Vtičnik je uporabljen samo za varen prenos dokumentov, saj morajo biti vhodne datoteke kriptirane z uporabo dogovorjenega javnega ključa.

### **3.2.3.2 Vtičniki za zajem in validacijo podatkov**

#### *XML VALIDATOR*

Vtičnik za iskanje vrednosti metapodatkov po XML dokumentih. Za iskanje uporablja XPATH poizvedeni jezik, ki omogoča administratorju aplikacije enostavno konfiguracijo.

Iskanje je omogočeno čez vse XML datoteke znotraj posredovanega dokumenta ter omogoča nastavitve lokacije iskanja (št. datoteke ali prva datoteka z najdenim vozliščem) in označitev dokumenta za izključitev z izvoznih modulov.

#### *REGEX VALIDATOR*

Vtičnik za iskanje vrednosti metapodatkov po vseh tekstovnih in PDF dokumentih. Vtičnik za prepoznano ali nastavljeno datoteko naloži tekstovno vsebino ter uporabi regularne izraze za iskanje ključne in iskane besede.

Dodatne možnosti:

- Določitev več ključnih besed za eno iskalno polje.
- Omejitev iskanja na določene dokumente in strani.
- Določitev zaporedne številke najdene ključne besede (npr.: Išči po 3ti najdeni iteraciji).

### *SUBSTRING VALIDATOR*

Vtičnik za iskanje formatiranih tekstovnih dokumentov. Za iskanje naloži vsebino tekstovnega dokumenta ter za vsak metapodatek bere nastavljeno vrstico, začetno pozicijo in dolžino.

Dodatne možnosti:

- Brisanje belih presledkov.
- Iskanje po vsebini že najdenega metapodatka.

### *CSV VALIDATOR*

Vtičnik za iskanje vrednosti metapodatkov po CSV datotekah različnih vrst. Za iskanje naloži vsebino datoteke ter za vsak metapodatek bere vrstico, določeno polje in ločitveni znak ali niz.

### *DATABASE VALIDATOR*

Vtičnik za uparjanje obstoječih vrednosti metapodatka z nastavljivo poizvedbo po poljubnih podatkovnih bazah. Kot nastavitev sprejema povezovalni niz, ki ga vtičnik zaradi možne vsebine gesla kriptira ter poizvedbo, ki povezuje polja metapodatkov z ciljno podatkovno bazo.

Primer: `Select Ime_Polja AS Ime_Metapodatka where Ime_Polja2 = @Ime_Metapodatka2`

### *DATE FORMATER*

Vtičnik za formatiranje datumskega polja. V nastavitve za vsak željen metapodatek po prioriteti vpišemo vhodni format (npr »YYYYMMDD«) ter ciljni format z možnostjo preverjanja pravilnosti datumske vrednosti.

Dodatne možnosti:

- Pri vhodnem formatu vpišemo X za neznan znak ali za neznana ločil.

- Pri izhodne formatu lahko vpišemo katerikoli znak za ločilo ali vsebino (vpisano z {} oklepaji).
- Možnost vpisa knjižnice za mesece, privzeto so že vpisani dolgi in kratki format za vsebino slovenskih in angleških.

### **3.2.3.3 Vtičniki za pretvorbo**

#### *PDF/A CONVERTER*

Vtičnik za pretvorbo besedilnih dokumentov v PDF/A obliko zapisa. Omogoča nastavljljive mere robov ter že obstoječo TIFF ali PDF predlogo za pisanje.

#### *TIF CONVERTER*

Vtičnik omogoča pretvorbo besedilnih dokumentov (TXT, XML in MHT) v rastrsko sliko TIF. Omogoča nastavljljive mere robov ter že obstoječo TIFF predlogo za pisanje.

#### *ZIP COMPRESOR*

Vtičnik omogoča stiskanje posamičnih datotek ali celotnega dokumenta v en ZIP zabojnik.

### **3.2.3.4 Vtičniki za izvoz**

#### *CSV/XML FILE RELEASE*

Nastavljljiv izhodni vtičnik za kreacijo poljubne XML ali CSV datoteke na podlagi celotnega paketa in/ali posamičnega dokumenta. Pri kreaciji CSV datoteke lahko poljubno nastavimo zaporedje metapodatkov, pri XML datoteki pa lahko določimo celotno predlogo XML datoteke z označbo, kjer naj bodo vpisane vrednosti in imena metapodatkov.

#### *ZIP RELEASE*

V uporabi skupaj z CSV/XML. Release dodaja možnost stiskanja celotnega paketa v eno ZIP datoteko.

# 4 Primeri praktične uporabe

## 4.1 Preproste obdelave

Za preproste obdelave običajno jemljemo tip dokumentacije, pri katerih ni potrebno opraviti velikega zajema metapodatkov oziroma pripraviti pretvorbe.

Dokumenti za preproste obdelave že pridejo ločeni v logične dokumente, tako, da tu prvi korak separacije izpustimo.

Po večini se gre tu za bolj preprost zajem metapodatkov in uvoz v ciljni sistem, le redko pa se opravlja še kakšna pretvorba.

Primeri preprostih obdelav:

- Elektronska plačila (interni sistemi).  
Dokumentacija pride z dogovorjenim imenovanjem datotek, ki ga uporabimo za uparjanje ostalih metapodatkov z Database Validatorjem. Opcijsko nam lahko posredujejo tudi datumsko polje v imenu datoteke, ki je ločen z posebnim znakom. To polje zajamemo z CSV validatorjem, formatiramo pa ga z Date Formater-jem.
- Digitalizirana dokumentacija.  
Dokumentacija, ki je že bila digitalizirana pri naročniku in opremljena z črtno kodo. Vrednost črtne kode pridobimo z imena datoteke, ki ga uporabimo za uvoz v ciljno dokumentacijo.
- Prejeti računi (digitalizirani).  
Enako kot pri ostali digitalizirani dokumentaciji. V imenu že pridobimo črtno kodo v imenu datoteke. Vrednost črtne kode uporabimo za uparjanje ostalih vrednosti z ciljnega ERP sistema z uparjanjem ali priklopom na njihov vmesnik.
- Izvozna dokumentacija (parčki XML in PDF/A ali TIF).  
XML in PDF pari pridejo z enakim imenovanjem ter se s tem označuje navezava datotek. V ciljni sistem uvozimo sam PDF/A ali TIF, XML pa služi za pridobitev metapodatkov.
- Dokumentacija plač.  
V imenu datotek se po dogovoru vpišejo že vsi metapodatki potrebni za zajem ter jih ločujemo z dogovorjenim znakom. Vrednost potem prebiramo z vtičnikom CSV validator (npr.; Datum\_IDDelavca\_NazivDelavca.txt).



- Dokumentacija izdanih računov.

Tekstovne dokumentacija pri kateri prebiramo glavo datoteke za vse ključne podatke, ki so vedno na enakem mestu. Za branje je uporabljen vtičnik SubString Validator.

## **4.2 Primeri zahtevnih obdelav**

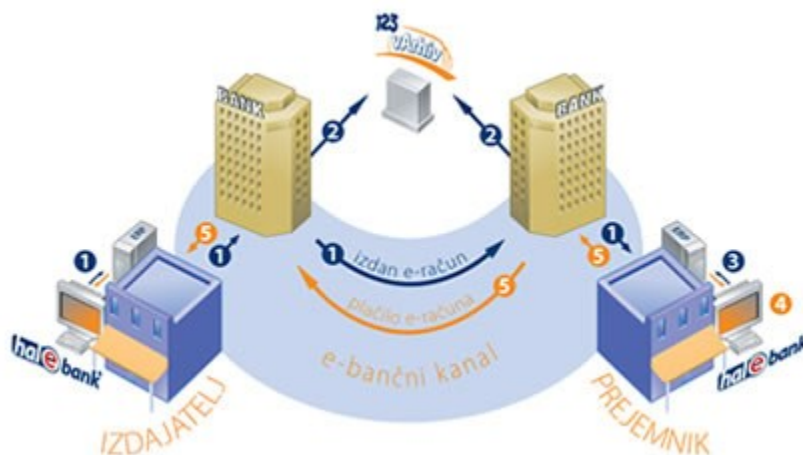
### **4.2.1 Prejeti elektronski računi**

GZS je s projektom e-SLOG v letih od 2001 do 2006 uspela povezati interese in strokovnjake iz več kot 100 podjetij in jih združiti pri pripravi in uveljavljanju enotnih slovenskih priporočil, ki podjetjem omogočajo elektronsko poslovanje. Rezultat projekta so standardi za elektronske dokumente, naročilnico, dobavnico in račun in več tehnoloških priporočil za povezovanje podjetij ter priporočila za uporabo elektronskega podpisa in arhiviranje elektronskih dokumentov. V letu 2010 je GZS skupaj z ZBS in ostalimi deležniki pričela z aktivnostmi za nadaljevanje vzdrževanja, nadgradnje in uveljavljanje teh standardov. [8]

Pripravljen standard se je v zadnjih letih že kar dodobra uveljavil, saj opazimo vedno več pošiljateljev e-računov. Za prejem elektronskih računov se moramo na njih prijaviti (za vsakega dobavitelja) ter jih prejemamo v svojo elektronsko banko, kar nam omogoča tudi o lažje in hitrejšje plačevanje. Pojavila pa se je težava pri procesu likvidacije in odobritve, ker so bili procesi in arhiv pripravljeni samo za obdelavo in prenos digitaliziranih računov.

Vhodni elektronski računi morajo biti z dogovorjenim standardom že posredovani v obliki za dolgoročno hrambo, zato nam tu ni potrebno skrbeti dodatno pretvorbo formata zapisa.

Spodnja slika prikazuje posredovanje računa in plačila med partnerji.



Slika 4.2.1.1: Prikaz posredovanja računa in plačila med partnerji (www.halcom.si)

*Vhodne datoteke:*

- XML Ovojnica računa
  - Vsebuje podatke za prenos računa med bankami pošiljatelja in prejemnika.
  - Vsebuje zapis o prilogah računa (vključno z posredovanim E-Slogom)
  - XML Ovojnica že vsebuje večino izmed ključnih metapodatkov, ki so potrebni za zajem dokumenta.
  - Ovojnica računa je potrebna zaradi prenosa med bankami

Ključni podatki vključeni v ovojnici:

- Pošiljatelj (Naziv, naslov, davčna številka, BIC koda banke in IBAN)
- Prejemnik (Naziv, naslov, davčna številka, BIC koda banke in IBAN)
- Številka računa
- Strukturirana referenca
- Končni znesek računa (brez ostalih postavk)
- E-Slog računa (trenutna verzija 1.6).  
Račun v XML obliki po specifikaciji E-Sloga , ki so jo pripravili ZBS, GZS in SETCE (pripravi dokumentacijo).  
Vsebuje vse podatke o računu (vključno z postavkami, obračunanimi davki ter datumi računa), pošiljatelju in prejemniku.  
Pošiljatelj lahko v glavo datoteke doda še XSLT datoteko za vizualizacijo dokumenta, ki omogoča končnim uporabnikom lažji pregled računa.

Tu lahko pridobimo ostale podatke, ki so pomembni za proces ali pa so dodani na željo končnih uporabnikov za potrebe lažje likvidacije in iskanja po arhivu.

- Račun v PDF/A obliki (Opcijsko).

V kolikor je posredovan račun kot PDF dokument mora biti vedno že pretvorjen v PDF/A pod standard. Pošiljatelj ga posreduje opcijsko in po-navadi vključuje enako vizualizacijo kot E-slog računa.

- Priloge računa (opcijske dodatne specifikacije ali obrazložitve)

Dovoljene priloge: PDF/A in XML oblika zapisa.

#### *Priprava in zajem dokumentov*

Naročnik mora na dogovorjeno lokacijo urediti odlaganje e-računov z ovojnico (glede na tip elektronske banke to lahko uredijo tudi avtomatsko.)

Aplikacija z dogovorjene lokacije ločuje ovojnice in priloge računa ter jih porazdeli v vsebinske dokumente. Z identifikacijo ovojnice računa, ki je glavna datoteka računa poišče še priloge (Eslog,PDF/A računa,...) ter jih dodeli pripadajočim dokumentom.

Poleg že nastavljenih metapodatkov za zajem v podatkovno strukturo za prenos doda novo polje, ki označuje številko dokumenta v ovojnici, za potrebe pravilnega branja pri zajemu metapodatkov.

Za separacijo in zajem je uporabljen vtičnik EInvoice Separator.

#### *Zajem metapodatkov*

Zajem metapodatkov uredimo z vtičnikom XML Validator. Administrator vanj vpiše nastavitve XPATH vrednosti z ovojnice ali E-Sloga računa, ki bodo uporabljene za zajem metapodatkov. Spodnji tabeli prikazujeta najbolj tipične vrednosti, ki jih prebiramo iz dokumenta (Po verziji E-Slog 1.6) .

<b>Pogosto uporabljene XPATH iskalne vrednosti za iskanje po E-Slog 1.6</b>	
<b>Naziv Metapodatka</b>	<b>XPath iskalna vrednost</b>
Številka računa	/IzdaniRacunEnostavni/Racun[1]/GlavaRacuna[1]/StevilkaRacuna[1]
Datum računa	/IzdaniRacunEnostavni/Racun[1]/DatumiRacuna[VrstaDatuma='137']/DatumRacuna.
Datum zapadlosti	/IzdaniRacunEnostavni/Racun[1]/PlacilniPogoji[1]/PlacilniRoki[VrstaDatumaPlacilnegaRoka='13']/Datum
Datum odpreme	/IzdaniRacunEnostavni/Racun[1]/DatumiRacuna[VrstaDatuma='35']/DatumRacuna

Sklic	/IzdaniRacunEnostavni/Racun[1]/PovzetekZneskovRacuna/SklicZaPlacilo[SklicPlacila='PQ']/StevilkaSklica[1]
-------	--

Tabela 4.2.1.1: Pogosto uporabljene XPath iskalne vrednosti za iskanje po E-Slog 1.6

<b>Pogosto uporabljene XPATH iskalne vrednosti za iskanje po ovojnici E-Računa</b>	
<b>Naziv Metapodatka</b>	<b>XPath iskalna vrednost</b>
Naziv pošiljatelja	/ns:package/ns:envelope/ns:sender/ns:name
Davčna pošiljatelja	/ns:package/ns:envelope/ns:sender/ns:sender_identifier
Datum zapadlosti	/ns:package/ns:envelope/ns:payment_data/ns:amount
Valuta	/ns:package/ns:envelope/ns:payment_data/ns:currency

Tabela 4.2.1.2: Pogosto uporabljene XPath iskalne vrednosti za iskanje po ovojnici E-računa

V tabeli »ns« predstavlja imenski prostor, ki je vpisan v nastavitvah.

#### *Uvoz v zaledni sistem*

V veliko primerih se tu izkaže dodatna zahteva za delovanje aplikacije, saj je potrebno dokumente uvoziti v dokumentni, arhivski in BPM sistem.

Ob dogovoru za delovanje aplikacije naročnik preda specifikacijo vmesnika za prenos zajetih dokumentov. Tu moramo običajno zaradi novih, tujih ali lastnih sistemov urediti nov prilagojen vtičnik, ki bo iz aplikacije prejel zajete dokumente in jih posredoval v ciljni sistem naročnika.

Dobra praksa priprave kaže, da je vedno potrebno te vtičnike graditi na splošen način, da bi bila možnost ponovne uporabe v enakem ciljnem sistemu. Prav tako pa je z dogovorom potrebno doseči, da se iz vmesnika poročajo statusi o uspešnosti prenosa.

## **4.2.2 PDF izpiski**

Potrebo po tovrstni obdelavi smo prvič zaznali, ko se je naročnik hotel izogniti digitalizaciji dokumentov, ki se niso spreminjali od njihovega tiska. To so bili dokumenti, ki jih ni bilo

potrebno podpisovati, ker se je šlo za obvestila, izpiske, opomine in podobno masovno pošto. Naročniki so tu po navadi urejali dodatne kopije za lastne potrebe arhiva, da so imeli omenjene izpiske zavedene, prav tako pa se je digitalizacija za elektronski arhiv izkazala za časovno potratno.

Prikazana rešitev le tega je bila, da se datoteke, ki jih posredujejo tiskarjem posebej odlagajo še v elektronski zajem, kar je pomenilo preskok dveh stroškovno in časovno potratnih procesov (dodatna kopija in digitalizacija).

#### *Vhodne datoteke*

PDF datoteke, ki imajo vključen eden ali več logičnih dokumentov. Ločevanje se opravi z spremembo ključnih vrednosti na dokumentu, ki so bile predhodno dogovorjene z naročnikom(npr.: Sprememba davčne ali IBAN številke).

#### *Priprava in zajem dokumentov*

Naročnik lahko PDF datoteke pripravi na različne načine. Pri večjih naročnikih je bila to samo dodatna kopija elektronskega izvirnika, ki se posreduje tiskarjem. Pri manjših pa so se posluževali uparjalnih orodij znotraj MS Office okolja (Mail Merge), tako, da so na dogovorjeno predlogo izpolnjevali ključna in vsebinska polja (ime priimek, zadeva ipd...).

Kot dodatni dogovor skušamo doseči dogovor o »skritem tekstu«. Skriti tekst imenujemo zapisano vsebino v levem zgornjem kotu vsakega novega dokumenta (znotraj PDF datoteke), ki je obarvano belo, tako, da se na natisnjenih dokumentih ne bo videl. Z tem dogovorom lahko zagotovimo višjo kvaliteto branja in hitrejše delovanje, saj aplikacija ne potrebuje brati celotnega dokumenta za najdbo pravih metapodatkov.

V beli tekst zapisujemo vrednosti z ključno besedo ter vrednostjo ( vsaka v novo vrstico).

Npr.:

Naziv: Marko Žankar

Davčna Številka: 12345678

V tem primeru bi za ključni metapodatek označili davčno številko, ki bi služila za ločevanje strani datoteke v svoje logične dokumente.

### *Zajem metapodatkov*

Zajem metapodatkov uredimo z vtičnikom ReGex Validator. Vanj administrator vpiše na iskalne vrednosti (npr.: Naziv) ter regularno vrednost, ki predstavlja vrednost metapodatka.

Spodnja tabela predstavlja vrednosti nekaterih najbolj pogosto uporabljenih regularnih izrazov za iskanje vrednosti.

<b>Pogosto uporabljeni regularni izrazi</b>	
<b>Tip podatka</b>	<b>Regularni izraz</b>
Davčna številka	<code>( SI\d{8} )   ( \d{8} )</code>
IBAN vrednost	<code>[a-zA-Z]{2}[0-9]{2}[a-zA-Z0-9]{4}[0-9]{7}([a-zA-Z0-9?){0,16}</code>
EMŠO	<code>\d{15}</code>
Datumska vrednost (DD.MM.YYYY)	<code>\d{1,2}\\.\\d{1,2}\\.\\d{4}</code>
Znesek	<code>(\d{1,3}(?:\.\d{3})*(?:,\d{2})?\$) (?:^\d{1,3}(?:,\d{3})*(?:\.\d{2})?)</code>

Tabela 4.2.2.1: Vrednosti najbolj pogosto uporabljenih regularnih izrazov za iskanje vrednosti.

### *Pretvorba*

Ob pretvorbi že imamo ločene dokumente, ki pa jih moramo pretvoriti v PDF/A obliko. V redkih primerih je imel naročnik željo za TIFF format datotek, zaradi skladnosti z ostalo dokumentacijo h kateri se je pripenjal opomin.

Za ustrezno pretvorbo je z naročnikom potrebna uskladitev posredovanih PDF dokumentov, ker teh dokumente ni vedno mogoče pretvoriti v PDF/A podstandard.

Kot dodatno opcijo jim omogočamo združevanje z naročnikovo predlogo dokumenta (predloga z logotipom, glavo in nogo naročnika), s katero je dosežen cilj enakosti dokumenta z natisnjenim.

### *Uvoz v zaledni sistem*

Glede na masovno količino posredovane pošte z opomini in izpiski, se izvoz oziroma uvoz opravlja paketno z uporabo vnaprej dogovorjenih standardov. Običajno se tu uredi CSV datoteka z vrstičnim zapisom metapodatkov za dokument, pri kateri je prvi zapis vrednosti

številka dokumenta na katero se vrstica zapisa nanaša. Možno je opraviti tudi pare z XML datoteko, ki hranijo vrednost o posredovani PDF/A datoteki.

Ker se gre tu po večini za izpiske, opomine, ipd..., je zelo redka praksa, da bi bila potrebo po uvozu v kakšen dodaten zaledni sistem (ERP ali BPM).

### **4.2.3 Obdelava in priprava izdanih računov**

Za pripravo izdanih računov iz tekstovnega dokumenta, ki ga lahko izda zaledni sistem naročnika, je želeni rezultat predstavil naročnik. Želja naročnika je bila pripraviti dokumente za tisk, jih sočasno arhivirati ter posredovani v naročnikov interni sistem za elektronsko obveščanje o posredovanju računa.

Naročnik se je želel izogniti ročnemu delu, ker so vse podatke že imeli pripravljene ter je njihov sistem omogočal strukturiran izvoz tekstovnih datotek. Z naročnikom smo tako uskladili predlogo z njihovim logotipom v PDF/A obliki ter določili vsebino, ki se bo na njej prikazovala.

#### *Vhodne datoteke*

Strukturirane tekstovne datoteke. Pričetek vsake strani je označen z posebno oznako. Prav tako pa je označen pričetek vsakega novega dokumenta z nizom interne številke računa.

#### *Priprava in zajem dokumentov*

Priprava na strani naročnika je zajemal samo ureditev avtomatskega ali ročnega odlaganja datotek po dogovorjeni strukturi v vhodno mapo programske rešitve.

Za separacijo in pripravo logičnih dokumentih tu skrbi vtičnik Text separator, ki na podlagi ključnih besed pripravi nove tekstovne dokumente, ki jih bo posredoval v kasnejši zajem metapodatkov.

#### *Zajem metapodatkov*

Glede na vhodne strukturirane dokumente je zajem relativno enostaven. Dogovorjeni so želeni metapodatki, vsebino pa zajamemo z vtičnikom SubString Validator, saj imamo natančno opredeljeno mesto in dolžino vseh metapodatkov.

Ob kakršnikoli spremembi je naročnik dolžan obvestiti pripravljavca, da ustrezno popravi in testira novo konfiguracijo

### *Pretvorba*

Ključni del tovrstne obdelave je predstavljala pretvorba tekstovnega dokumenta v želeno obliko PDF/A formata, ki vključuje elektronski podpis. Največje delo in težavnost nam je predstavljalo pripenjanje vsebine tekstovnega dokumenta na predlogo, ki se je morala natančno ujemati. Ob ureditvi in dokončni potrditvi je naročnik izdal še certifikat s katerim mora, biti PDF dokument podpisan.

### *Uvoz v zaledni sistem*

Uvoz v zaledni sistem je pri tovrstni obdelavi skoraj vedno priprava prilagojenega vtičnika, ki bo omogočala izvoz dokumentov v ciljni sistem, ki bo naročniku omogočal najhitrejšo distribucijo in posredovanje dokumentov v tisk.



## 5 Sklepne ugotovitve

Z postopno vpeljavo programske rešitve smo dosegli vse zadane cilje, katere smo opredelili pred pripravo programske rešitve. Osnova za postavitev ciljev je bila izvedena analiza obstoječega stanja.

Z izdelavo programske rešitve in njeno uporabo smo omogočili, da priprava nove obdelave za zajem dokumentov za vsa nova naročila, ne zahteva sodelovanje sistemskih analitikov ali programerjev, kar občutno skrajša čas priprave obdelave. Pripravo, administracijo in nadzor sedaj lahko opravljajo tehnologi, kar nam omogoča boljšo konkurenčnost in odzivnost ter razbremenitev razpoložljivih virov.

Prenos programske opreme v produkcijo je pokazal, da je bila rešitev dobro zamišljena, saj je pokrila vse uporabljane funkcionalnosti predhodno uporabljenih rešitev, jih celo nadgradila ter združila na skupni imenovalci. Tu se je izkazalo, da nam je uspelo doseči tudi cilj zmanjševanja stroškov, kar se kaže kot zmanjšanje stroškov vzdrževanja in porabe delovnih virov.

Poenotenje je omogočilo lažjo vsebinsko skrb nad zajemom, saj smo zmanjšali posebnosti posamičnih obdelav in s tem omogočili skupini ljudi, ki sodeluje pri pripravi novih obdelav, da se hitreje seznani z potekom obdelave.

Analiza in priprava rešitve je zahtevala temeljito preučitev zakonodaje, s katero smo nadgradili znanje, ki ga potrebujemo, da bomo lažje opravljali delo na svojem področju v podjetju, ki se ukvarja z zelo specifično dejavnostjo, to je zajem in elektronska hramba gradiva v elektronski obliki. Poleg tega pa tudi nove izkušnje s področja razvoja programske opreme.

## 6 Literatura

- [1] AIIM, »What is Document Management(DMS)« .Dostopno na:  
<http://www.aiim.org/What-is-Document-Management>
- [2] Arhiv Republike Slovenije, Enotne tehnološke zahteve za zajem in hrambo gradiva v digitalni obliki, Različica 2.1, 2013. Dostopno na:
- [http://www.arhiv.gov.si/fileadmin/arhiv.gov.si/pageuploads/E-ARHIVI/ETZ\\_2\\_1/ETZ\\_-\\_1\\_del\\_razlicica\\_2.1\\_-\\_koncna.pdf](http://www.arhiv.gov.si/fileadmin/arhiv.gov.si/pageuploads/E-ARHIVI/ETZ_2_1/ETZ_-_1_del_razlicica_2.1_-_koncna.pdf)
  - [http://www.arhiv.gov.si/fileadmin/arhiv.gov.si/pageuploads/E-ARHIVI/ETZ\\_2\\_1/ETZ\\_-\\_II.\\_del\\_razlicica\\_2.1\\_-\\_koncna.doc.pdf](http://www.arhiv.gov.si/fileadmin/arhiv.gov.si/pageuploads/E-ARHIVI/ETZ_2_1/ETZ_-_II._del_razlicica_2.1_-_koncna.doc.pdf)
  - [http://www.arhiv.gov.si/fileadmin/arhiv.gov.si/pageuploads/E-ARHIVI/ETZ\\_2\\_1/ETZ\\_2.1\\_III.del\\_-\\_koncna.pdf](http://www.arhiv.gov.si/fileadmin/arhiv.gov.si/pageuploads/E-ARHIVI/ETZ_2_1/ETZ_2.1_III.del_-_koncna.pdf)
- [3] Arhiv republike Slovenije, Enotne tehnološke zahteve, Različica 1.0, 2006. Dostopno na:  
<http://www.arhiv.gov.si/fileadmin/arhiv.gov.si/pageuploads/E-ARHIVI/obrazci/ETZ.pdf>
- [4] Adobe Systems incorporated, TIFF, 1992. Dostopno na:  
<http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>
- [5] Aquaforest Limited, »TIFF versus PDF for Document Storage«, 2008. Dostopno na:  
[http://www.aquaforest.com/en/tiff\\_versus\\_pdf.asp](http://www.aquaforest.com/en/tiff_versus_pdf.asp)
- [6] Callas J., Finney H. (PGP Corporation), Donnerhacke L.,(IKS GmbH) Shaw D. in Thayer R., »OpenPGP Message Format«, 2007 . Dostopno na:  
<http://www.ietf.org/rfc/rfc4880.txt>
- [7] Evropska komisija, MOREQ2:Zbirka zahtev za delo z elektronskimi zapisi, 2008. Dostopno na:  
<http://ec.europa.eu/>
- [8] Gospodarska zbornica Slovenije, Združenje bank Slovenije, SETCEE, Projekt e-Slog (Priporočila za uporabo standarda GZS e-SLOG 1.5 za enostavni račun), 2010. Dostopno na:  
[http://www.gzs.si/e-poslovanje/dokumentacija/eSLOG\\_1-5\\_EnostavniRacun\\_prirocnik.pdf](http://www.gzs.si/e-poslovanje/dokumentacija/eSLOG_1-5_EnostavniRacun_prirocnik.pdf)
- [9] Microsoft Corporation, C# Programming Guide Dostopno na:  
<http://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>
- [10] North Dakota Information Technology Department, »Enterprise Report Management«. Dostopno na:  
<http://www.nd.gov/itd/services/enterprise-report-management>

- [11]Pkware Inc., ZIP File Format Specification, 2012. Dostopno na:  
<http://www.pkware.com/documents/casestudies/APPNOTE.TXT>
- [12]Republika Slovenija, Zakon o varstvu dokumentarnega in arhivskega gradiva ter arhivih (ZVDAGA). Uradni list RS, št. 30/2006
- [13]Republika Slovenija, Uredba o varstvu dokumentarnega in arhivskega gradiva (UVDAG). Uradni list RS, št. 86/2006
- [14]Rising Jim, »Sashimi Waterfall Software Development Process«, 2009. Dostopno na:  
<http://www.managedmayhem.com/2009/05/06/sashimi-waterfall-software-development-process/>
- [15]SAA – Society of American Archivists, »A Glossary of Archival and Records Terminology«, 2012. Dostopno na:  
<http://www2.archivists.org/glossary/>
- [16]Vasilescu Ramona, »PDF/A standard for long term archiving«, Tibiscus University, Timisoara, 2009. Dostopno na:  
<http://arxiv.org/pdf/0906.0867.pdf>
- [17]W3schools, »Introduction to XML«. Dostopno na:  
[http://www.w3schools.com/xml/xml\\_what\\_is.asp](http://www.w3schools.com/xml/xml_what_is.asp)
- [18]Weitzer Aleks, Primerjava in vrednotenje procesov razvoja programske opreme (Magistersko delo), Ljubljana, 2009. Dostopno na:  
[http://eprints.fri.uni-lj.si/1017/1/Aleks\\_Weitzer\\_magi.pdf](http://eprints.fri.uni-lj.si/1017/1/Aleks_Weitzer_magi.pdf)



Št. naloge: 00507 / 2013  
Datum: 15.4.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MARKO ŽANKAR**

Naslov: **RAZVOJ APLIKACIJE ZA ARHIVIRANJE DOKUMENTOV**  
**THE DEVELOPMENT OF APPLICATION FOR DOCUMENT ARCHIVING**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Osnovni namen orodij širšega področja arhiviranja dokumentov je omogočiti obdelavo gradiv: zajem dokumentov iz zunanjih informacijskih sistemov, morebitno proženje delovnih procesov na podlagi vsebine dokumentov ter po zaključku procesov arhiviranje teh dokumentov v sistem za elektronsko hrambo dokumentov. Razvoj in vzdrževanje takšnih namenskih aplikacij predstavlja visoke stroške vzdrževanja in administriranja, saj so praviloma v takšni verigi tri do štiri aplikacije. Zasnujte, izdelajte načrt in razvite aplikacijo, ki bo obsegala funkcionalnosti celotne zgoraj omenjene verige. Aplikacija naj uporabniku omogoča enostavno pripravo nove obdelave gradiva brez programiranja. Aplikacija naj uporablja programske vtičnike, ki so razdeljenimi v kategorije separacije, validacije, pretvorbe in izhodnih modulov. Arhitektura aplikacije naj omogoča, da se novi vtičniki lahko razvijajo ločeno od osnovne aplikacije.

Mentor:

doc. dr. Rok Rupnik

Dekan:

prof. dr. Nikolaj Zimic

